Fetch Take home assessment Part 2

For this exercise, I connected the jupyter notebook to the PostgresQL database server to load the 3 datasets into the PostgresQL environment.

Checks on join of transactions and products -

- The number of rows increased after performing a left join on transactions with the products dataset which indicates there must've been a duplication of records.
- On checking, I came across two RECEIPT_IDs that duplicated after join operation
    - RECEIPT_ID "2f984bec-0243-475c-9e1e-ed9933f70fd1" and "b59e828a-2040-4188-aa9f-70b53cdc854d" associated with BARCODE "4003207"
    - BARCODE "4003207" is associated with two values in CATEGORY_2 and CATEGORY_3.

	CATEGORY_1 text	CATEGORY_2 text	CATEGORY_3 text	CATEGORY_4 text	MANUFACTURER text	BRAND text	BARCODE double precision
1	Snacks	Nuts & Seeds	Peanuts	[null]	MARS WRIGLEY	M&M'S	4003207
2	Snacks	Candy	Chocolate Candy	[null]	MARS WRIGLEY	M&M'S	4003207

Assumptions made -

- Due to the lack of any information/context on the above BARCODE, I have chosen the second record as the true record for "4003207" BARCODE mentioned above and removed the first record from the dataset to avoid any duplication of records. The second record seems more appropriate for M&M'S as it is a candy.
- FINAL_SALE value of ' ' (blank) replaced with 0 for aggregations
- FINAL_QUANTITY value of 'zero' replaced with 0 for aggregations

```
clean_trans_df.loc[clean_trans_df['FINAL_SALE'] == ' ', 'FINAL_SALE']= '0.00'
clean_trans_df.loc[clean_trans_df['FINAL_QUANTITY'] == 'zero', 'FINAL_QUANTITY']= '0.00'
```

Code for checks on duplication of records after join operation-

```
#Count of records per receipt id
WITH receipt_count_table AS (
SELECT "RECEIPT_ID", COUNT(*) AS count1
FROM transactions
GROUP BY "RECEIPT_ID"),

#Count of records per receipt id after join of transactions and products
join_receipt_count_table AS (
SELECT t."RECEIPT_ID", COUNT(*) AS count2
FROM
transactions t
LEFT JOIN products p
ON t."BARCODE" = p."BARCODE"
GROUP BY t."RECEIPT_ID"),

#Joining the the two above CTEs and performing a check on counts at RECEIPT_ID level
receipt_count_check_table AS (
SELECT *, (count1 - count2) AS countcheck
FROM
receipt_count_table LEFT JOIN join_receipt_count_table
ON receipt_count_table."RECEIPT_ID" = join_receipt_count_table."RECEIPT_ID")
#Extracting ids of receipt that duplicated and see the associated BARCODE
```

Submitted by Raghav Goel

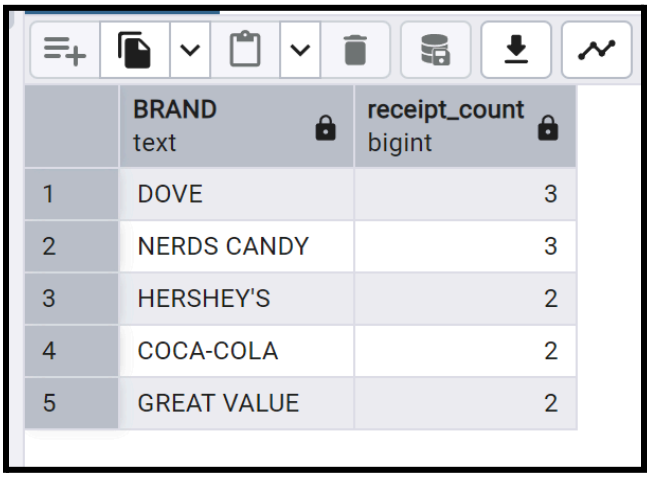
```
SELECT* FROM receipt_count_check_table
WHERE countcheck <>0;

#Query to remove the row not required for the '4003207' BARCODE
DELETE FROM products
WHERE "BARCODE" = '4003207' AND "CATEGORY_2" = 'Nuts & Seeds';
```

Close ended -

1. **What are the top 5 brands by receipts scanned among users 21 and over?** Filtered out "null" brand value from this question. The top 5 brands are Dove, Nerds Candy, Hershey's, Coca-Cola and Great Value

Query - The below code uses extract function and case statement to take users 21 and over based on the current date (Dynamic in nature). **Please note that there are more brands with 2 receipts scanned other than the ones shown below in the screenshot. Currently the result shows top 5 just based on how the SQL environment prints them sorted on number of receipts scanned but there should be more criterias that can be added to determine true "top" 5 amongst the brands that have 2 receipts scanned.**

```
SELECT p."BRAND", COUNT(DISTINCT "RECEIPT_ID") AS receipt_count  #selecting Brand and distinct count of receipt ID
FROM transactions t
LEFT JOIN users u ON t."USER_ID" = u."ID"  #performing a left join on users dataset
LEFT JOIN products p ON t."BARCODE" = p."BARCODE"  #performing a left join on products dataset
WHERE p."BRAND" IS NOT NULL  #filtering out null brand values
AND (EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM "BIRTH_DATE"))
   - CASE
      WHEN EXTRACT(MONTH FROM CURRENT_DATE) < EXTRACT(MONTH FROM "BIRTH_DATE")
      OR (EXTRACT(MONTH FROM CURRENT_DATE) = EXTRACT(MONTH FROM "BIRTH_DATE")
      AND EXTRACT(DAY FROM CURRENT_DATE) < EXTRACT(DAY FROM "BIRTH_DATE"))
      THEN 1 ELSE 0
   END >= 21  #calculating exact age based on current date and filtering for 21 and over
GROUP BY p."BRAND"
ORDER BY receipt_count DESC
LIMIT 5;
```



	BRAND text	receipt_count bigint
1	DOVE	3
2	NERDS CANDY	3
3	HERSHEY'S	2
4	COCA-COLA	2
5	GREAT VALUE	2

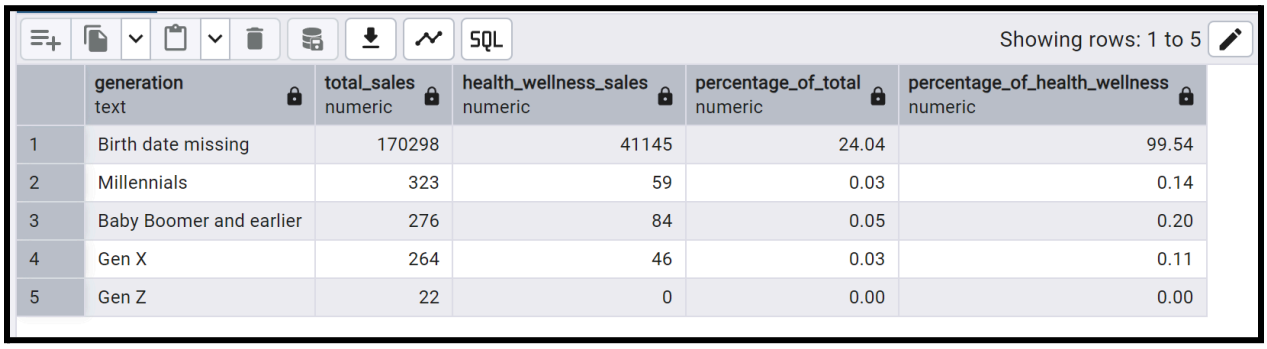
Submitted by Raghav Goel

2. **What is the percentage of sales in the Health & Wellness category by generation?** Defined generation based on birth_date. 24% of Total Sales are associated with records that do not have a birth date specified and 99% of sales within Health & Wellness are associated with records that do not have a birth date specified.

Query -

```
SELECT
CASE #creating generation column based on age range
        WHEN u.age IS NULL THEN 'Birth date missing'
        WHEN u.age<=12 THEN 'Gen Alpha'
        WHEN u.age>12 AND u.age<=28 THEN 'Gen Z'
        WHEN u.age>28 AND u.age<=44 THEN 'Millennials'
        WHEN u.age>44 AND u.age<=60 THEN 'Gen X'
        WHEN u.age>60 THEN 'Baby Boomer and earlier'
END AS generation,
ROUND(SUM(t."FINAL_SALE")::NUMERIC,0) AS total_sales, #total sales in transactions dataset
ROUND(SUM(CASE WHEN p."CATEGORY_1" = 'Health & Wellness' THEN t."FINAL_SALE" ELSE 0
END)::NUMERIC,0) AS health_wellness_sales, #total health & wellness category sales
ROUND(
(SUM(CASE
                WHEN p."CATEGORY_1" = 'Health & Wellness' THEN t."FINAL_SALE"
                ELSE 0
        END)*100)::NUMERIC
        /(SELECT SUM(t."FINAL_SALE")::NUMERIC
        FROM transactions t
        LEFT JOIN products p ON t."BARCODE" = p."BARCODE")
,2)AS percentage_of_total, #health & wellness sales as a % of total sales
ROUND(
(SUM(CASE
                WHEN p."CATEGORY_1" = 'Health & Wellness' THEN t."FINAL_SALE"
                ELSE 0
        END)*100)::NUMERIC
        /(SELECT SUM(t."FINAL_SALE")::NUMERIC
        FROM transactions t
        LEFT JOIN products p ON t."BARCODE" = p."BARCODE"
        WHERE p."CATEGORY_1"='Health & Wellness')
,2) AS percentage_of_health_wellness #%of total health & wellness sales
FROM
transactions t
LEFT JOIN (SELECT *, (EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM "BIRTH_DATE"))
  - CASE
    WHEN EXTRACT(MONTH FROM CURRENT_DATE) < EXTRACT(MONTH FROM "BIRTH_DATE")
    OR (EXTRACT(MONTH FROM CURRENT_DATE) = EXTRACT(MONTH FROM "BIRTH_DATE")
    AND EXTRACT(DAY FROM CURRENT_DATE) < EXTRACT(DAY FROM "BIRTH_DATE"))
    THEN 1 ELSE 0
    END AS age
FROM users) u #using subquery to add an Age column in users database for generation grouping
ON t."USER_ID" = u."ID" #performing a left join on users dataset
LEFT JOIN products p ON t."BARCODE" = p."BARCODE" #performing a left join on products dataset
GROUP BY generation ORDER BY total_sales DESC; #showing the metrics by generation
```

Result -



Showing rows: 1 to 5

	generation text	total_sales numeric	health_wellness_sales numeric	percentage_of_total numeric	percentage_of_health_wellness numeric
1	Birth date missing	170298	41145	24.04	99.54
2	Millennials	323	59	0.03	0.14
3	Baby Boomer and earlier	276	84	0.05	0.20
4	Gen X	264	46	0.03	0.11
5	Gen Z	22	0	0.00	0.00

Open ended -

3. **Which is the leading brand in the Dips & Salsa Category?** I have considered the leading brand in the Dips & Salsa Category based on the number of receipts scanned, number of stores the brand's product is bought in, total sales & quantity. The brand 'TOSTITOS' seems to be the leading brand across all metrics.

Query -

```
SELECT p."BRAND",
COUNT(DISTINCT t."RECEIPT_ID") AS receipt_count, #distinct count of receipts scanned
ROUND(SUM(t."FINAL_SALE")::NUMERIC,0) AS total_sales, #total sales
COUNT(DISTINCT t."STORE_NAME") AS total_stores, #number of stores
SUM(t."FINAL_QUANTITY") AS total_quantity #total quantity
FROM transactions t
LEFT JOIN products p ON t."BARCODE" = p."BARCODE"
WHERE p."CATEGORY_2" = 'Dips & Salsa' AND p."BRAND" IS NOT NULL #filtering for Dips & Salsa category and filtering out null brand
GROUP BY p."BRAND"
ORDER BY receipt_count DESC; #currently ordered by receipt_count but when ordered by other metrics, TOSTITOS still stays the leader
```

Result-

	BRAND text	receipt_count bigint	total_sales numeric	total_stores bigint	total_quantity double precision
1	TOSTITOS	36	261	16	60
2	PACE	24	119	8	38
3	FRITOS	19	92	9	33
4	DEAN'S DAIRY DIP	17	68	4	22
5	MARKETSIDE	16	103	1	23